# FORWARD-BACKWARD CONVOLUTIONAL RECURRENT NEURAL NETWORKS AND TAG-CONDITIONED CONVOLUTIONAL NEURAL NETWORKS FOR WEAKLY LABELED SEMI-SUPERVISED SOUND EVENT DETECTION

*Janek Ebbers, Reinhold Haeb-Umbach*

Paderborn University, Germany
{ebbers, haeb}@nt.upb.de

## ABSTRACT

In this paper we present our system for the *detection and classification of acoustic scenes and events (DCASE) 2020 Challenge Task 4: Sound event detection and separation in domestic environments*. We introduce two new models: the forward-backward convolutional recurrent neural network (FBCRNN) and the tag-conditioned convolutional neural network (CNN). The FBCRNN employs two recurrent neural network (RNN) classifiers sharing the same CNN for preprocessing. With one RNN processing a recording in forward direction and the other in backward direction, the two networks are trained to jointly predict audio tags, i.e., weak labels, at each time step within a recording, given that at each time step they have jointly processed the whole recording. The proposed training encourages the classifiers to tag events as soon as possible. Therefore, after training, the networks can be applied to shorter audio segments of, e.g., 200 ms, allowing sound event detection (SED). Further, we propose a tag-conditioned CNN to complement SED. It is trained to predict strong labels while using (predicted) tags, i.e., weak labels, as additional input. For training pseudo strong labels from a FBCRNN ensemble are used. The presented system scored the fourth and third place in the systems and teams rankings, respectively. Subsequent improvements allow our system to even outperform the challenge baseline and winner systems in average by, respectively, 18.0 % and 2.2 % event-based $F_1$-score on the validation set. Source code is publicly available at `https://github.com/fgnt/pb_sed`.

***Index Terms—*** audio tagging, event detection, weak labels

## 1. INTRODUCTION

Environmental sound recognition is recently gaining increased interest from both academia and industry. Plenty of applications potentially benefit from reliable sound recognition such as ambient assisted living, autonomous driving and environmental monitoring. Depending on the application different acoustic information is required. While acoustic scene classification aims at classifying the acoustic environment, audio tagging and sound event detection (SED) aim at recognizing specific sounds [1]. The latter two differ in the provided level of detail with audio tagging only indicating the presence of a sound in a recording of, e.g., 10 s, and SED aiming at on- and offset detection within a certain collar of, e.g., 200 ms.

With deep neural networks dominating state-of-the-art (SOTA) sound recognition, training requires labeled data, which, however, is expensive and time-consuming particularly with strong labels, i.e., when event on- and offsets have to be annotated. Due to the additional effort for strong labeling, large scale databases like Google's AudioSet [2] usually only provide weak labels which only indicate presence/absence of certain sounds within audio recordings. Therefore, the first challenge in SED is to learn to predict event on- and offsets despite the weak audio tagging labels provided during training. Further, semi-supervised learning tries to only use few labeled data while exploiting unlabeled data to improve performance.

Driven by the annual detection and classification of acoustic scenes and events (DCASE) challenges [3, 4, 5], the SOTA in weakly labeled semi-supervised SED has progressed rapidly over the last years. Several approaches have been proposed for weakly labeled SED [6, 7, 8, 9] most of which are based on multiple instance learning pooling functions [10]. Most recent SOTA approaches, e.g., [9, 11, 12, 13], rely on neural attention. To perform audio tagging a neural network learns to attend to the time range where the sound event is active. Afterwards the network can be used to locate sound events in time although no strong labels have been used during training. Semi-supervised SED is dominated by teacher student approaches [14, 15], where the teacher and student networks are jointly trained employing an additional loss for consistency between their predictions on unlabeled data.

In this paper we present our system for the *DCASE 2020 Challenge Task 4: Sound event detection and separation in domestic environments* [16] tackling weakly labeled semi-supervised SED with multi-label classification, i.e., multiple events can be active at a time. It aims at SED of ten different sound classes in real audio recordings from a domestic environment.

We present a weakly labeled SED approach based on two new models: the forward-backward convolutional recurrent neural network (FBCRNN) and tag-conditioned convolutional neural network (CNN). The FBCRNN employs a shared CNN and two recurrent neural networks (RNNs), one processing the input audio signal in forward direction and the other in backward direction. The RNNs are encouraged to tag events as soon as possible by training them to jointly predict audio tags at each time step within a recording, given that at each time step the two RNNs have jointly processed the whole recording. After training, the networks can also be used for SED by applying them to short audio segments of, e.g., 200 ms. As a complement, tag-conditioned CNNs are trained to predict strong labels when getting tags as additional input. Here, pseudo strong labels from a FBCRNN ensemble are used for training.

It is shown that the proposed approach is highly competitive and outperforms current SOTA approaches. A rather naive pseudo labeling [17] of unlabeled data is shown to improve performance. We hypothesize that a more sophisticated approach to semi-supervised learning, such as a mean-teacher approach [14], may even further improve performance in the future.

The rest of the paper is structured as follows. Sec. 2 explains our feature extraction and data augmentation. Then the FBCRNN and tag-conditioned CNN models are introduced in Sec. 3 and Sec. 4, respectively. Finally, in Sec. 5 experiments are presented and conclusions are drawn in Sec. 6.

This work has been supported by Deutsche Forschungsgemeinschaft under contract no. HA 3455/15-1 within the Research Unit FOR 2457. Computational resources were provided by the Paderborn Center for Parallel Computing.

## 2. FEATURE EXTRACTION AND DATA AUGMENTATION

Our system's input $\mathbf{X}$ is a 128 dimensional log-mel spectrogram using a short-time Fourier transform (STFT) with a hop-size of 20 ms, a frame length of 60 ms and a sampling rate of 16 kHz. Waveforms are initially normalized to be within the range of -1 and 1: $x(t) = s(t)/\max(|s(t)|)$. Each mel bin of the log-mel spectrogram is globally normalized to zero mean and unit variance.

During training various data augmentation techniques are used, namely random scaling, mixup, frequency warping, blurring, time masking, frequency masking and random noise. Random scaling and mixup [18] are performed on the waveform similar as in [19] by shifting and superposing signals as follows:

$$x'_i(t) = \sum_{j=0}^{J_i-1} \lambda_j x_j(t - \tau_j)$$

with the time shift $\tau_j$ being uniformly sampled such that $x'_i(t)$ is not longer than the maximum mixture length $T_{\text{max}}$, the mixture weights $\lambda_j$ being sampled from $\text{LogNormal}(0, 1)$ and the distribution of the number of mixture components $J_i$ being

$$\Pr(J_i) = \begin{cases} 1 - p_{\text{mix}} & J = 1, \\ p_{\text{mix}} & J = 2, \\ 0 & \text{else.} \end{cases}$$

The mixup probability $p_{\text{mix}}$ and $T_{\text{max}}$ are hyper-parameters. Note the difference to original mixup [18] as we do not apply an interpolation to the signals but a superposition. Therefore, we also do not interpolate the targets but combine them into a single multi-hot target vector. Frequency warping and time- and frequency masking are performed exactly as in [19]. Blurring is performed with a gaussian blur kernel of size $5 \times 5$ where the standard deviation of the kernel is randomly sampled from $\text{Exp}(0.5)$. Finally, random gaussian noise is added to the log-mel spectrogram with the noise power being randomly sampled from $\text{Uniform}(0, 0.2)$.

## 3. FORWARD-BACKWARD CONVOLUTIONAL RECURRENT NEURAL NETWORK

To allow for SED learned from weak labels, we aim to make a RNN classifier (either processing an input signal in forward or backward direction) to immediately tag an event at the frame it appears. If, however, a RNN is only trained to predict tags after it has processed the whole signal, as in [19], there is no reason it should output tags immediately. More likely, it learns to gather information over long segments and not make hasty decisions. To encourage the model to make immediate predictions, one might think to train a RNN to predict event tags after each frame. If an event, however, occurs only at the end of an audio-clip, the model would be trained to tag the event before it has seen it. Here, we therefore suggest a joint training of a forward and backward RNN such that at each frame at least one of the two RNNs correctly tags an active event and both do not tag inactive events. We hypothesize that forward and backward classifiers are able to jointly perform a tagging at each time frame as the forward classifier has seen all sound events between first and current frame while the backward classifier has seen all sound events between current and last frame. This way we encourage the classifiers to tag sound events as soon as they appear. Fig. 1 illustrates the prediction behavior of the forward and backward classifiers. With a shared CNN as pre-processing we refer to the proposed model as forward-backward convolutional recurrent neural network (FBCRNN).

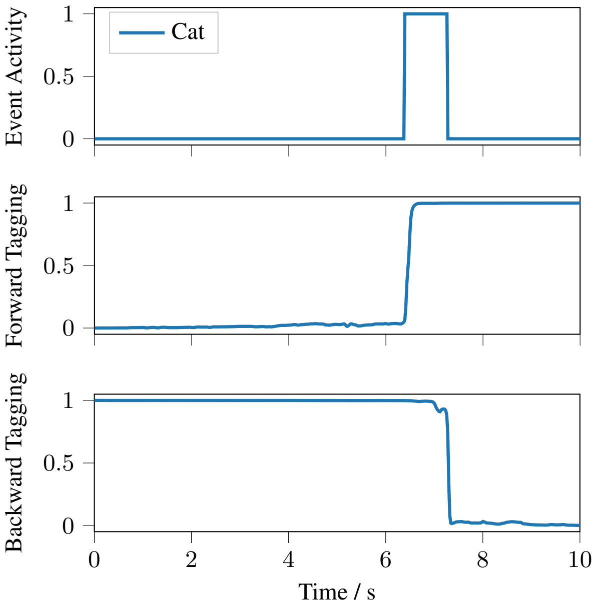

Figure 1: Illustration of forward and backward tagging.

Table 1: CNN architecture as in [19] but without temporal pooling. Each ConvXd uses a kernel size of three and a stride of one and includes BatchNorm [20] and ReLU.

| Block | output shape |
|---|---|
| LogMel(128) | $1{\times}128{\times}N$ |
| GlobalNorm | $1{\times}128{\times}N$ |
| 2×Conv2d(16) | $16{\times}128{\times}N$ |
| Pool2d(2×1) | $16{\times}64{\times}N$ |
| 2×Conv2d(32) | $32{\times}64{\times}N$ |
| Pool2d(2×1) | $32{\times}32{\times}N$ |
| 2×Conv2d(64) | $64{\times}32{\times}N$ |
| Pool2d(2×1) | $64{\times}16{\times}N$ |
| 2×Conv2d(128) | $128{\times}16{\times}N$ |
| Pool2d(2×1) | $128{\times}8{\times}N$ |
| Conv2d(256) | $256{\times}8{\times}N$ |
| Pool2d(2×1) | $256{\times}4{\times}N$ |
| Reshape | $1024{\times}N$ |
| 3×Conv1d(256) | $256{\times}N$ |

An input $\mathbf{X}$ is forwarded through the CNN $\mathbf{H} = f_{\text{cnn}}(\mathbf{X})$. The CNN architecture is shown in Tab. 1. We then perform recurrent forward tagging $\mathbf{Y}^{\text{fwd}} = f_{\text{rnn}}^{\text{fwd}}(\mathbf{H})$ and backward tagging $\overleftarrow{\mathbf{Y}}^{\text{bwd}} = f_{\text{rnn}}^{\text{bwd}}(\overleftarrow{\mathbf{H}})$, with $\overleftarrow{\mathbf{H}}$ denoting time flipped $\mathbf{H}$. Note the difference to a bidirectional RNN as here the forward and backward RNNs do not exchange hidden representations. The RNN architecture is shown in Tab. 2.

During training tag predictions are computed at each time frame by taking the maximum of the forward and backward prediction $\mathbf{Y}^{\text{tag}} = \max(\mathbf{Y}^{\text{fwd}}, \mathbf{Y}^{\text{bwd}})$ with $\max(\cdot)$ denoting point wise maximum operation.

The training criterion is the binary cross entropy between tag predictions $\mathbf{y}_n^{\text{tag}}$ and the multi-hot target vector $\mathbf{z}^{\text{tag}}$:

$$L(\mathbf{y}_n^{\text{tag}}, \mathbf{z}^{\text{tag}}) = -\sum_{k=0}^{K-1} \left( z_k^{\text{tag}} \log(y_{n,k}^{\text{tag}}) + (1 - z_k^{\text{tag}}) \log(1 - y_{n,k}^{\text{tag}}) \right)$$

with $K = 10$ denoting the number of target event classes.

Table 2: Recurrent classifier architecture as in [19].

| Block | output shape |
|---|---|
| 2×GRU(256) | $256{\times}N$ |
| $\text{fc}_{\text{ReLU}}(256)$ | $256{\times}N$ |
| $\text{fc}_{\text{Sigmoid}}(K)$ | $K{\times}N$ |

At inference time audio tag predictions are obtained as the average of the forward and backward predictions when they have processed the whole signal:

$$\hat{\mathbf{y}}^{\text{tag}} = f_{\text{tag}}(\mathbf{X}) = (\mathbf{Y}^{\text{fwd}}[N-1] + \mathbf{Y}^{\text{bwd}}[0])/2$$

with $N$ denoting the number of frames in $\mathbf{X}$. Event-specific tagging thresholds $\alpha_k$ are used to get binary tag predictions

$$\hat{z}_k^{\text{tag}} = \left[\hat{y}_k^{\text{tag}} > \alpha_k\right] = \begin{cases} 1, & \hat{y}_k^{\text{tag}} > \alpha_k, \\ 0, & \text{else.} \end{cases}$$

SED is achieved by applying tagging to a small context around each frame: $\hat{\mathbf{y}}_n^{\text{sed}} = f_{\text{tag}}(\mathbf{X}_{n-C_k:n+C_k}) \cdot \hat{\mathbf{z}}^{\text{tag}}$ with $n$ denoting the frame index and $C_k$ denoting an event-specific one-sided context length. Again applying event-specific detection thresholds $\beta_k$ yields binary predictions $\hat{z}_{n,k}^{\text{sed}} = \left[\hat{y}_{n,k}^{\text{sed}} > \beta_k\right]$. Finally median filtering with an event-specific filter size $M_k$ is applied to get the final SED.

## 4. TAG-CONDITIONED CONVOLUTIONAL NEURAL NETWORK

Second, we propose a tag-conditioned CNN to perform SED by directly predicting strong labels. This model can be understood as a two-stage approach, where tags are predicted in a first stage, here by the FBCRNN, and given the tag predictions a subsequent model, here a CNN, predicts strong labels. We hypothesize that the CNN can do better event detection if it is aware of the active events within a recording, i.e. when it has to predict $\Pr(\text{event active in frame}|\text{event active in recording})$ rather than $\Pr(\text{event active in frame})$.

The CNN architecture $\hat{\mathbf{y}}_n^{\text{sed}} = f_{\text{sed}}(\mathbf{X}_{n-R:n+R}, \mathbf{z}^{\text{tag}})$ is similar as in Tab. 1 with $R = 13$ being the one-sided receptive field of the CNN. Hence, the overall receptive field is $2R+1 = 27$ frames corresponding to 580 ms. In contrast to Tab. 1 the last Conv1d layer is outputting $K = 10$ scores here, one for each event class. Further, in addition to the (augmented) log-mel input spectrogram, this second-stage CNN is conditioned on audio tags by concatenating a multi-hot tag encoding $\mathbf{z}^{\text{tag}}$ to each time-frequency bin along the channel dimension of the log-mel spectrogram as well as to the hidden representation between the reshape operation and the first Conv1d layer.

At training time (pseudo) weak labels $\mathbf{z}^{\text{tag}}$ are used as conditioning. The training criterion is the frame-wise binary cross-entropy between predictions $\hat{\mathbf{y}}_n^{\text{sed}}$ and (pseudo) strong labels $\mathbf{z}_n^{\text{sed}}$:

$$L(\hat{\mathbf{y}}_n^{\text{sed}}, \mathbf{z}_n^{\text{sed}}) = -\sum_{k=0}^{K-1} \left( z_{n,k}^{\text{sed}} \log(\hat{y}_{n,k}^{\text{sed}}) + (1-z_{n,k}^{\text{sed}}) \log(1-\hat{y}_{n,k}^{\text{sed}}) \right)$$

which is averaged over all frames in a mini-batch. If no strong labels are available, pseudo strong labels from some other weakly trained SED model, such as the FBCRNN, can be used for training.

At inference time the CNN is conditioned on tag predictions $\hat{\mathbf{z}}^{\text{tag}}$ from the FBCRNN. We apply event-wise decision thresholds $\beta_k$, as before, to get binary frame predictions $\hat{z}_{n,k}^{\text{sed}} = \left[\hat{y}_{n,k}^{\text{sed}} > \beta_k\right]$ and subsequent median filtering with an event-specific filter size $M_k$.

## 5. EXPERIMENTS

Experiments are performed using the DESED database used in the fourth task of the *DCASE* 2019 and *DCASE* 2020 Challenges [5, 16]. The database features three training sets for SED, namely a small weakly labeled data set of 1578 real audio recordings, 2584 synthetic soundscapes with strong labels and a larger set of 14412 unlabeled real audio recordings. To adjust the percentage of the different data sets in training, in each epoch recordings from the weakly labeled and synthetic data sets are presented 10 and 2 times, respectively, resulting in a data distribution of 75.3 % weakly labeled and 24.7 % synthetic data. In experiments using pseudo labeled recordings from the unlabeled data set, these recordings are presented once in each epoch resulting in a data distribution of 44.6 % weakly labeled, 14.6 % synthetic and 40.8 % unlabeled data. For the synthetic data, we further perform on-the-fly reverberation of individual sound events.

All trainings are performed for 40000 update steps with checkpointing and validation every 1000-th update step. Mini-batches of size $B$ are randomly sampled from the training data such that no signal in the mini batch is padded by more than 5% and each mini-batch includes at least $\lfloor B/3 \rfloor$ examples from the weakly labeled data set. Adam [21] is used for optimization with gradient clipping at a threshold of 20 and a learning rate ramp up to $5{\cdot}10^{-4}$ over the first 1000 update steps and a learning rate reduction to $1{\cdot}10^{-4}$ for update steps $>$15000. While for FBCRNNs the checkpoint with the best macro-averaged audio tagging $\text{F}_1$-score on the validation set is adopted as the final model, for CNN models the checkpoint with the best macro-averaged frame-based $\text{F}_1$-score is chosen.

Reported audio tagging performance is the macro-averaged $\text{F}_1$-score and reported SED performance is the macro-averaged event-based $\text{F}_1$-score [22] using an onset collar of 200 ms and an offset collar of 200 ms or 20 % of event duration if the duration is $>$1 s. If not stated otherwise, the hyper-parameters $\alpha_k$, $\beta_k$, $C_k \in \{5, 10, 15, 20\}$ and $M_k \in \{11, 21, 31, 41, 51\}$ are tuned to give best performance on the validation set. Since the labels of the DCASE 2020 Task 4 evaluation set (eval-2020) are not public yet, performance is primarily evaluated on the publicly available *DCASE 2019 Task 4 youtube evaluation set* (yt-eval-2019) [23, 5]. If not stated otherwise, each experiment is repeated five times from which we report the mean and standard deviation.

Ensembles combine four independently trained models by averaging their model outputs $\hat{\mathbf{y}}$ if not stated otherwise.

First, we evaluate the single-model tagging and detection performance of the proposed FBCRNN. In particular the effectiveness of the proposed forward-backward approach for weakly labeled learning is investigated as well as the usefulness of pseudo labeling the unlabeled data for semi-supervised learning. Tab. 3 compares the following models:

- $\text{CRNN}^{\text{last-only}}_{\text{no-pseudo}}$: CRNN w/ only forward tagging where loss is only computed at the last frame of an audio recording (see [19]) and trained w/o unlabeled data set,
- $\text{CRNN}_{\text{no-pseudo}}$: CRNN w/ only forward tagging where loss is computed at each frame of an audio recording and trained w/o unlabeled data set,
- $\text{FBCRNN}_{\text{no-pseudo}}$: FBCRNN trained w/o unlabeled data set,
- $\text{FBCRNN}_{\text{submitted}}$: FBCRNN used in our submission trained w/ a heuristical on-the-fly pseudo labeling [24] (means and standard deviations computed over only four models here),
- FBCRNN: FBCRNN trained w/ unlabeled data set after weakly pseudo labeled by an $\text{FBCRNN}_{\text{no-pseudo}}$ ensemble.

Table 3: Single-model tagging and detection performance of CRNNs in terms of macro-averaged (event-based) F$_1$-scores[%].

| Ensemble | validation | | yt-eval-2019 | |
|---|---|---|---|---|
| | Tagging | Detection | Tagging | Detection |
| $\text{CRNN}^{\text{last-only}}_{\text{no-pseudo}}$ | 81.8±0.2 | 25.5±0.6 | 80.9±0.7 | 18.1±0.5 |
| $\text{CRNN}_{\text{no-pseudo}}$ | 79.6±0.5 | 34.0±0.7 | 78.7±0.4 | 30.9±1.4 |
| $\text{FBCRNN}_{\text{no-pseudo}}$ | 82.6±0.1 | 40.7±1.3 | 81.8±0.5 | 40.3±1.9 |
| $\text{FBCRNN}_{\text{submitted}}$ | 82.3±0.4 | 42.0±0.2 | 81.8±0.5 | 41.3±1.1 |
| FBCRNN | 84.8±0.2 | 46.4±0.5 | 84.1±1.2 | 47.4±1.1 |

The models are trained using a mini-batch size of $B = 16$ and mixup with $p_{\text{mix}} = 2/3$ and $T_{\text{max}} = 15\,\text{s}$.

It can be observed that the $\text{CRNN}^{\text{last-only}}_{\text{no-pseudo}}$ gives good audio tagging performance but fails to perform SED. This confirms our hypothesis from Sec. 3 that without a frame-wise loss the model does not learn to tag sounds in short contexts. Using a frame-wise loss with the forward-only CRNN ($\text{CRNN}_{\text{no-pseudo}}$) improves SED to some extent. However, training the model to tag events at frames, where it may not have seen the events yet, limits performance as can be seen by the deterioration of tagging. Using the proposed forward backward tagging approach ($\text{FBCRNN}_{\text{no-pseudo}}$) significantly improves SED performance. Interestingly it also improves audio tagging performance over the $\text{CRNN}^{\text{last-only}}_{\text{no-pseudo}}$ model. Finally, the on-the-fly pseudo labeling ($\text{FBCRNN}_{\text{submitted}}$), that was used in our submitted system, only slightly improves detection performance over the model trained without unlabeled data ($\text{FBCRNN}_{\text{no-pseudo}}$). However, pseudo labeling using a $\text{FBCRNN}_{\text{no-pseudo}}$ ensemble and training a new FBCRNN by also leveraging pseudo labeled data, improves tagging and detection performance significantly.

Next the effectiveness of the tag-conditioned CNN is evaluated. For that we compare CNNs with and without tag conditioning. For training strong pseudo labels for the weakly and unlabeled data sets are obtained by an FBCRNN ensemble using hyper parameters giving best frame-based $F_1$-scores on the validation set. The FBCRNN ensemble also provides the tags for conditioning. The models are trained using a mini-batch size of $B = 24$ and mixup with $p_{\text{mix}} = 1/2$ and $T_{\text{max}} = 12\,\text{s}$. Tab. 4 compares SED performance of the two models. First, it can be noted that both CNNs improve detection performance over the FBCRNN. This suggests that pseudo strong label training may improve performance in general. Comparing the CNN models to each other, it can be seen that the tag conditioning improves average performance from 50.1% to 53.4% event-based $F_1$.score on the evaluation set.

Finally, ensemble performance is compared to challenge baseline and winner systems. Tab. 5 lists detection performance for the following systems:

- Baseline: baseline system 2020 [16],
- Winner2019: winner system 2019 [15],
- Winner2020: winner system 2020 [25],
- $\text{Hybrid}_{\text{submitted}}$: Our submitted system consisting of the four $\text{FBCRNN}_{\text{submitted}}$ and four tag conditioned CNNs trained on pseudo strong labels from $\text{FBCRNN}_{\text{submitted}}$ and where for hyper-parameter tuning only $C_k \in \{5, 10\}$ and $M_k \in \{21, 41\}$ have been considered [24],
- $\text{Hybrid}^{*}_{\text{submitted}}$: Models from $\text{Hybrid}_{\text{submitted}}$ with the more extensive hyper-parameter tuning,
- FBCRNN: FBCRNN ensemble,
- CNN: CNN ensemble w/ tag conditioning,
- Hybrid: Combination of FBCRNN and CNN ensembles,

Ensembling gives on yt-eval-2019 an average gain of 2.4% and 0.7% event-based $F_1$-score over single-model FBCRNNs and CNNs (Tab. 3 and Tab. 4), respectively. It can be noted that the CNN ensemble clearly outperforms the baseline, winner and our submitted system. Combining the four FBCRNNs and four CNNs from individual ensembles to a larger Hybrid ensemble of eight submodels yields another slight improvement of 0.5% over the CNN-only ensemble. However, do note that the CNN-only detection is computationally much more efficient, as the FBCRNN detection relies on processing a context of length $1 + 2C$ at each frame for all $C \in \{C_k\}_{k=1}^{K}$. Comparing $\text{Hybrid}_{\text{submitted}}$ and $\text{Hybrid}^{*}_{\text{submitted}}$, which only differ in the hyper-parameters $\alpha_k$,$\beta_k$,$C_k$ and $M_k$, shows that the more extensive hyper-parameter tuning improves performance only slightly. Therefore, the improvement of the CNN and Hybrid ensembles over our submitted system comes mainly due to the improved pseudo-labeling resulting in better FBCRNN performance, as already seen in Tab. 3, and consequently in better pseudo strong labels for CNN training and finally in better overall performance. Tab. 6 shows event wise performance of our proposed ensembles on yt-eval-2019 with bold values highlighting the best ensemble.

Table 4: Single-model detection performance of CNN models in terms of macro-averaged event-based F$_1$-scores[%].

| Ensemble | validation | yt-eval-2019 |
|---|---|---|
| $\text{CNN}_{\text{no-cond.}}$ | 49.4±0.4 | 50.1±0.2 |
| CNN | 51.5±0.7 | 53.4±0.7 |

Table 5: Ensemble detection performance in terms of macro-averaged event-based F$_1$-scores[%].

| Ensemble | validation | yt-eval-2019 | eval-2020 |
|---|---|---|---|
| Baseline | 34.8 | - | 34.9 |
| Winner2019 | 45.3 | 47.7 | - |
| Winner2020 | 50.6 | - | 51.1 |
| $\text{Hybrid}_{\text{submitted}}$ | 48.3 | 50.8 | 47.2 |
| $\text{Hybrid}^{*}_{\text{submitted}}$ | 49.2 | 51.1 | - |
| FBCRNN | 48.3±0.4 | 49.9±1.1 | - |
| CNN | 52.1±0.5 | 54.1±0.7 | - |
| Hybrid | 52.8±0.6 | 54.6±0.5 | - |

Table 6: Event-wise detection performance on yt-eval-2019 in terms of event-based F$_1$-scores[%].

| Event | FBCRNN | CNN | Hybrid |
|---|---|---|---|
| Alarm bell ringing | 30.0±1.3 | **45.4±3.6** | 44.8±1.8 |
| Blender | 50.4±4.3 | 54.9±2.8 | **56.5±4.1** |
| Cat | 67.3±1.9 | **69.7±3.5** | 68.7±2.7 |
| Dishes | 31.2±1.8 | 29.5±2.9 | **32.1±3.9** |
| Dog | **48.3±1.6** | 44.1±2.5 | 48.2±3.3 |
| E. shaver/toothbrush | 44.6±1.8 | **52.5±3.0** | 49.2±6.4 |
| Frying | 58.5±1.0 | 61.4±1.1 | **61.7±1.2** |
| Running water | 39.7±4.1 | 45.7±2.7 | **46.6±2.1** |
| Speech | 60.8±1.4 | 64.0±0.8 | **64.1±1.1** |
| Vacuum cleaner | 68.4±1.1 | 73.5±1.1 | **73.8±1.8** |

## 6. CONCLUSIONS

In this paper we introduced the FBCRNN, a weakly labeled SED model trained to perform forward and backward tagging. The proposed training allows the model to also be applied to small segments of only a few hundred milliseconds enabling SED. On top we proposed a CNN, which is conditioned on audio tags, as a complementary SED model. Our system scored the fourth and third place in the systems and teams ranking, respectively, of the DCASE 2020 Challenge Task 4. Subsequent improvements allow our system to even outperform the baseline and winner systems in average by, respectively, 18.0 % and 2.2 % event-based $F_1$-score on the validation set.

## 7. REFERENCES

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational analysis of sound scenes and events*. Springer, 2018.

[2] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[3] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE2017 challenge setup: Tasks, datasets and baseline system," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, pp. 85–92.

[4] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 19–23.

[5] N. Turpault, R. Serizel, J. Salamon, and A. P. Shah, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 253–257.

[6] S. Adavanne and T. Virtanen, "Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, pp. 12–16.

[7] A. Shah, A. Kumar, A. G. Hauptmann, and B. Raj, "A closer look at weak label learning for audio events," *arXiv preprint arXiv:1804.09288*, 2018.

[8] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2180–2193, 2018.

[9] S.-Y. Chou, J.-S. R. Jang, and Y.-H. Yang, "Learning to recognize transient sound events using attentional supervision." in *IJCAI*, 2018, pp. 3336–3342.

[10] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 31–35.

[11] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, "Multi-level attention model for weakly supervised audio classification," *arXiv preprint arXiv:1803.02353*, 2018.

[12] L. Lin, X. Wang, H. Liu, and Y. Qian, "Specialized decision surface and disentangled feature for weakly-supervised polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1466–1478, 2020.

[13] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Weakly-supervised sound event detection with self-attention," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 66–70.

[14] L. JiaKai, "Mean teacher convolution system for dcase 2018 task 4," DCASE2018 Challenge, Tech. Rep., September 2018.

[15] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning convolution system for dcase 2019 task 4," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 134–138.

[16] N. Turpault and R. Serizel, "Training sound event detection on a heterogeneous dataset," 2020, working paper or preprint.

[17] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013.

[18] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[19] J. Ebbers and R. Hb-Umbach, "Convolutional recurrent neural network and data augmentation for audio tagging with noisy labels and minimal supervision," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 64–68.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[23] N. Turpault, R. Serizel, A. Shah, and J. Salamon, "Desed_public_eval," Dec. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3588172

[24] J. Ebbers and R. Haeb-Umbach, "Convolutional recurrent neural networks for weakly labeled semi-supervised sound event detection in domestic environments," Paderborn University, Paderborn, Germany, Tech. Rep., June 2020.

[25] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution-augmented transformer for semi-supervised sound event detection," Nagoya University, Japan, Tech. Rep., June 2020.